Milestone 1: Index Construction Report

Algorithms and Data Structures Developer Option

This report presents the analytics for the inverted index constructed using the **Algorithms and Data Structures Developer** approach. This option is required for CS and SE students.

Approach Specifications:
• **Corpus:** All ICS web pages (developer.zip) - ~56,000 pages
• **Index Storage:** File system (no databases) - multiple partial indexes merged
• **Memory:** Cannot hold entire index in memory - uses disk-based offloading
• **Offloading:** Index offloaded to disk at least 3 times during construction
• **Search Response Time:** Target ≤ 300ms (ideally ≤ 100ms)
• **Programming Level:** Advanced - efficient data structures and file access

Index Specifications:
• Tokens: All alphanumeric sequences
• Stop words: Not used (all words indexed)
• Stemming: Porter stemming algorithm
• Important words: Words in bold, headings (h1-h3), and titles are marked
• Term frequency: Calculated for each token in each document

Index Analytics

Metric	Value
Number of Indexed Documents	1,212
Number of Unique Tokens	13,126
Total Size of Index on Disk	14438.72 KB
Partial Indexes Created (Offloads)	4

■ **Requirement Verification:**
The indexer successfully created 4 partial indexes during construction, meeting the requirement of at least 3 offloads to disk.

Implementation Details

Disk-Based Indexing Architecture:
The indexer uses a memory-efficient approach that periodically offloads the in-memory index to disk as

partial index files. This ensures the index can be built for datasets of any size without running out of memory.

Processing Workflow:
1. Documents are processed and added to an in-memory index chunk
2. When memory limit is reached, the current index is saved as a partial index file
3. In-memory index is cleared and processing continues
4. Steps 1-3 repeat until all documents are processed
5. All partial indexes are loaded and merged into a single final index
6. Final merged index is saved to disk
7. Partial index files are cleaned up

File Organization:
• **inverted_index.json:** Final merged inverted index
• **doc_mapping.json:** URL to document ID mappings
• **partial_indexes/:** Temporary directory containing partial indexes during construction

Memory Management:
Memory usage is bounded by the chunk size (typically 300-5,000 documents per chunk). This ensures the indexer can handle very large datasets (56,000+ pages) without exceeding available memory. The chunk size is automatically calculated to ensure at least 3 offloads occur during construction.

Search Component Requirements (Milestone 2):
The search component must read postings from disk without loading the entire index into memory. This requires efficient disk-based lookups and data structures optimized for fast query response times.